Complexity of decision problems on totally rigid acyclic tree grammars

Sebastian Eberhard Gabriel Ebner Stefan Hetzl

DLT 2018
2018-09-11

TU Wien

Totally rigid acyclic tree grammars

Decision problems

Minimal cover

Conclusion

Totally rigid acyclic tree grammars

- Generate finite set of terms/trees
- TRATG is a tuple $G = (\mathrm{A}, N, \Sigma, P)$:
  - Start nonterminal $\mathrm{A} \in N$
  - Nonterminals N (arity 0)
  - Function symbols Σ
  - Productions $\mathrm{B} \to t$ where $B \in N$ and t a term
  - **acyclic**: $\mathrm{B}_1 \to t_1[\mathrm{B}_2], \ldots, \mathrm{B}_n \to t_n[\mathrm{B}_1]$ disallowed

Derivations

- $t[\mathrm{B}] \to t[s]$ where $\mathrm{B} \to s \in P$
- $\mathrm{A} \to^* t$

- **totally rigid**: at most one production per nonterminal
  - c.f. rigid tree automata (Jacquemard 2011)
  - choice of productions completely determines derived term

- $L(G) = \{t \mid \mathrm{A} \to^* t\} = \{\mathrm{B}_1[\mathrm{B}_1 \backslash t_1] \ldots [\mathrm{B}_n \backslash t_n] \mid \mathrm{B}_1 = \mathrm{A}, \forall i\, \mathrm{B}_i \to t_i \in P\}$

TRATG example

$$G = (\mathrm{A}, \{\mathrm{A}, \mathrm{B}\}, \{\mathsf{f}/2, \mathsf{g}/2, \mathsf{c}/0, \mathsf{d}/0\}, P)$$

$$P = \begin{cases} \mathrm{A} \to \mathsf{f}(\mathrm{B}, \mathrm{B}) \mid \mathsf{g}(\mathrm{B}, \mathrm{B}) \\ \mathrm{B} \to \mathsf{c} \mid \mathsf{d} \end{cases}$$

TRATG example

$$G = (\mathrm{A}, \{\mathrm{A}, \mathrm{B}\}, \{\mathsf{f}/2, \mathsf{g}/2, \mathsf{c}/0, \mathsf{d}/0\}, P)$$

$$P = \begin{cases} \mathrm{A} \to \mathsf{f}(\mathrm{B}, \mathrm{B}) \mid \mathsf{g}(\mathrm{B}, \mathrm{B}) \\ \mathrm{B} \to \mathsf{c} \mid \mathsf{d} \end{cases}$$

$$\mathrm{A} \to \mathsf{f}(\mathrm{B}, \mathrm{B}) \to \mathsf{f}(\mathsf{c}, \mathsf{c})$$

TRATG example

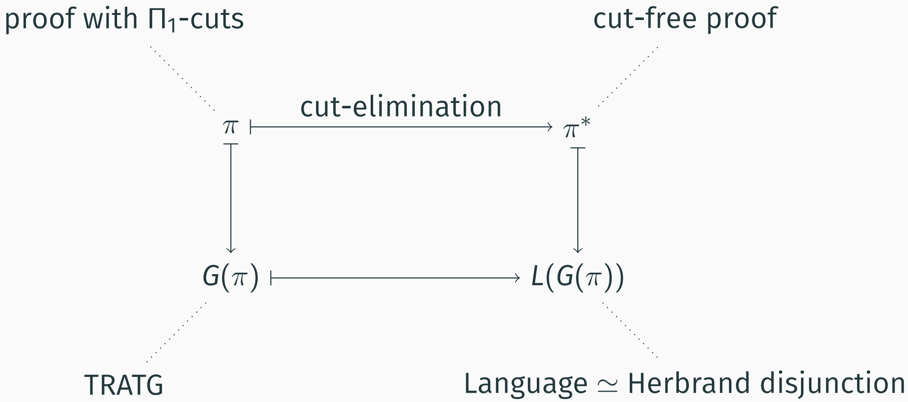
$$G = (\mathrm{A}, \{\mathrm{A}, \mathrm{B}\}, \{\mathsf{f}/2, \mathsf{g}/2, \mathsf{c}/0, \mathsf{d}/0\}, P)$$

$$P = \begin{cases} \mathrm{A} \to \mathsf{f}(\mathrm{B}, \mathrm{B}) \mid \mathsf{g}(\mathrm{B}, \mathrm{B}) \\ \mathrm{B} \to \mathsf{c} \mid \mathsf{d} \end{cases}$$

$$\mathrm{A} \to \mathsf{f}(\mathrm{B}, \mathrm{B}) \to \mathsf{f}(\mathsf{c}, \mathsf{c})$$

$$L(G) = \{\mathsf{f}(\mathsf{c}, \mathsf{c}), \mathsf{f}(\mathsf{d}, \mathsf{d}), \mathsf{g}(\mathsf{c}, \mathsf{c}), \mathsf{g}(\mathsf{d}, \mathsf{d})\}$$

Proof theory



Applications in proof theory

- nonterminal $\widehat{=}$ (quantifier in) Π_1-cut
- production $\widehat{=}$ quantifier inference
- generated term $\widehat{=}$ instance in Herbrand disjunction

- Lower bounds on compressibility using TRATGs translate to proofs (Eberhard, Hetzl 2018)
- Compression using small covering grammars $\rightarrow$ interesting lemmas (E, Hetzl, Leitsch, Reis, Weller 2018)

$\rightarrow$ Open source GAPT framework for proof theory: `https://logic.at/gapt`

Totally rigid acyclic tree grammars

Decision problems

Minimal cover

Conclusion

Membership

Problem (Membership)
Given a TRATG G and a term t, is $t \in L(G)$?

Membership

Problem (Membership)
Given a TRATG G and a term t, is $t \in L(G)$?

Claim: NP-complete.

- Derivations of t are w.l.o.g. polynomial in the size of t and G (dag-like!).

$$\mathrm{A} \to \mathrm{A}[\mathrm{A}\backslash s_1] \to \mathrm{A}[\mathrm{A}\backslash s_1][\mathrm{B}\backslash s_2] \to \ldots \to t$$

  Can check in polynomial time whether such a sequence of terms is a derivation of t in G.

- Hardness: next slide.

Membership (NP-hardness): encoding SAT

$L(\mathrm{Sat}_{n,m})$ = satisfiable 3-CNFs with n clauses and m variables:

$$
\begin{aligned}
\mathrm{A} &\to \mathtt{and}(\mathrm{Clause}_1, \ldots, \mathrm{Clause}_n) \\
\mathrm{Clause}_i &\to \mathtt{or}(\mathrm{True}_i, \mathrm{Any}_{i,1}, \mathrm{Any}_{i,2}) \\
\mathrm{Clause}_i &\to \mathtt{or}(\mathrm{Any}_{i,1}, \mathrm{True}_i, \mathrm{Any}_{i,2}) \\
\mathrm{Clause}_i &\to \mathtt{or}(\mathrm{Any}_{i,1}, \mathrm{Any}_{i,2}, \mathrm{True}_i) \\
\mathrm{Any}_{i,k} &\to \mathtt{x}_1 \mid \mathtt{neg}(\mathtt{x}_1) \mid \cdots \mid \mathtt{x}_m \mid \mathtt{neg}(\mathtt{x}_m) \mid \mathtt{false} \mid \mathtt{true} \\
\mathrm{True}_i &\to \mathrm{Value}_1 \mid \cdots \mid \mathrm{Value}_m \mid \mathtt{true} \\
\mathrm{Value}_j &\to \mathtt{x}_j \mid \mathtt{neg}(\mathtt{x}_j)
\end{aligned}
$$

Containment

Problem (Containment)
Given TRATGs G_1 and G_2, is $L(G_1) \subseteq L(G_2)$?

Problem (Containment)
Given TRATGs G_1 and G_2, is $L(G_1) \subseteq L(G_2)$?

Claim: Π_2^P-complete

- In Π_2^P: for every sequence of terms check if it is a derivation of a term t in G_1, and then if $t \in L(G_2)$.

Containment (Π_2^P-hardness)

- Determining the truth of a quantified Boolean formula $\forall y_1 \dots \forall y_k \exists x_1 \dots \exists x_m\, f$ is Π_2^P-complete. (f in 3-CNF)

- $f\sigma$ satisfiable for any $\sigma\colon \{y_1, \dots, y_k\} \to \{\mathtt{true}, \mathtt{false}\}$?
- $\{f\sigma \mid \sigma\colon \{y_1, \dots, y_k\} \to \{\mathtt{true}, \mathtt{false}\}\} \subseteq L(\mathrm{Sat}_{n,m})$?
- Left side is generated by a TRATG:

$$\mathrm{A} \to f[y_1 \backslash \mathrm{Y}_1, \dots, y_k \backslash \mathrm{Y}_k]$$
$$\mathrm{Y}_j \to \mathtt{true} \mid \mathtt{false}$$

Summary

$t \in L(G)$	NP-complete
$L(G_1) \subseteq L(G_2)$	Π_2^P-complete
$L(G_1) \cap L(G_2) = \emptyset$	coNP-complete
$L(G_1) = L(G_2)$	Π_2^P-complete

Totally rigid acyclic tree grammars

Decision problems

Minimal cover

Conclusion

Minimal cover

Problem (Minimal TRATG cover)

Given $k \geq 0$ and finite set of terms L,
is there a TRATG $G = (\mathrm{A}, N, \Sigma, P)$
such that $|P| \leq k$ and $L(G) \supseteq L$?

Minimal cover

Problem (Minimal regular cover)

Given $k \geq 0$ and finite set of words L,
is there a acyclic regular grammar $G = (\mathrm{A}, N, \Sigma, P)$
such that $|P| \leq k$ and $L(G) \supseteq L$?

Problem (Minimal regular n-cover)

Given $k \geq 0$ and finite set of words L,
is there a acyclic regular grammar $G = (\mathrm{A}, N, \Sigma, P)$
such that $|P| \leq k$ and $L(G) \supseteq L$, and $|N| \leq n$?

Complexity of minimal cover

hardness $\longleftarrow$

# nonterminals	terms	words
unbounded	?	?
bounded	NP-complete	NP-complete

membership $\longrightarrow$

- For $L(G) = L$, see talk by Gruber, Holzer, Wolfsteiner after lunch.

NP-completeness of minimal regular n-cover

Theorem
Minimal regular/TRATG n-cover is NP-complete ($n \geq 2$).

Proof.
NP-membership by reduction to membership. Hardness:

$$\begin{aligned} \text{SAT} &\leq_P \text{Minimal regular 2-cover-extension} \\ &\leq_P \text{Minimal regular 2-cover} \\ &\leq_P \text{Minimal regular 3-cover} \\ &\leq_P \dots \\ &\leq_P \text{Minimal regular } n\text{-cover} \end{aligned}$$

□

Totally rigid acyclic tree grammars

Decision problems

Minimal cover

Conclusion

Conclusion

- Membership, containment, disjointness, equivalence are hard
    - Because of equality constraints (due to total rigidity)
- Complexity of minimal cover remains unknown,
  even for acyclic regular (word) grammars

SAT $\leq_P$ Minimal regular 2-cover-extension

m clauses, n variables: $x_1, \ldots, x_n$.

$N = \{\mathrm{A}, \mathrm{B}\}$
$L(G) \supseteq \{s^{2n+1} o_{l,i} \mid i \leq m, l \leq 2n\}$
$P \supseteq \{\mathrm{B} \rightarrow s^{2n-2j} o_{l,i} \mid x_j \in C_i, l \leq 2n\}$
$P \supseteq \{\mathrm{B} \rightarrow s^{2n+1-2j} o_{l,i} \mid \neg x_j \in C_i, l \leq 2n\}$
$|P| \leq n + 2n \sum_i |C_i|$

x_i true:
$\mathrm{A} \rightarrow s^{2j}\mathrm{B} \rightarrow s^{2j} s^{2n+1-2j} o_{l,i}$

x_i false:
$\mathrm{A} \rightarrow s^{2j+1}\mathrm{B} \rightarrow s^{2j+1} s^{2n-2j} o_{l,i}$

$\rightarrow \quad I \models x_j$ iff G contains production $\mathrm{A} \rightarrow s^{2j}\mathrm{B}$